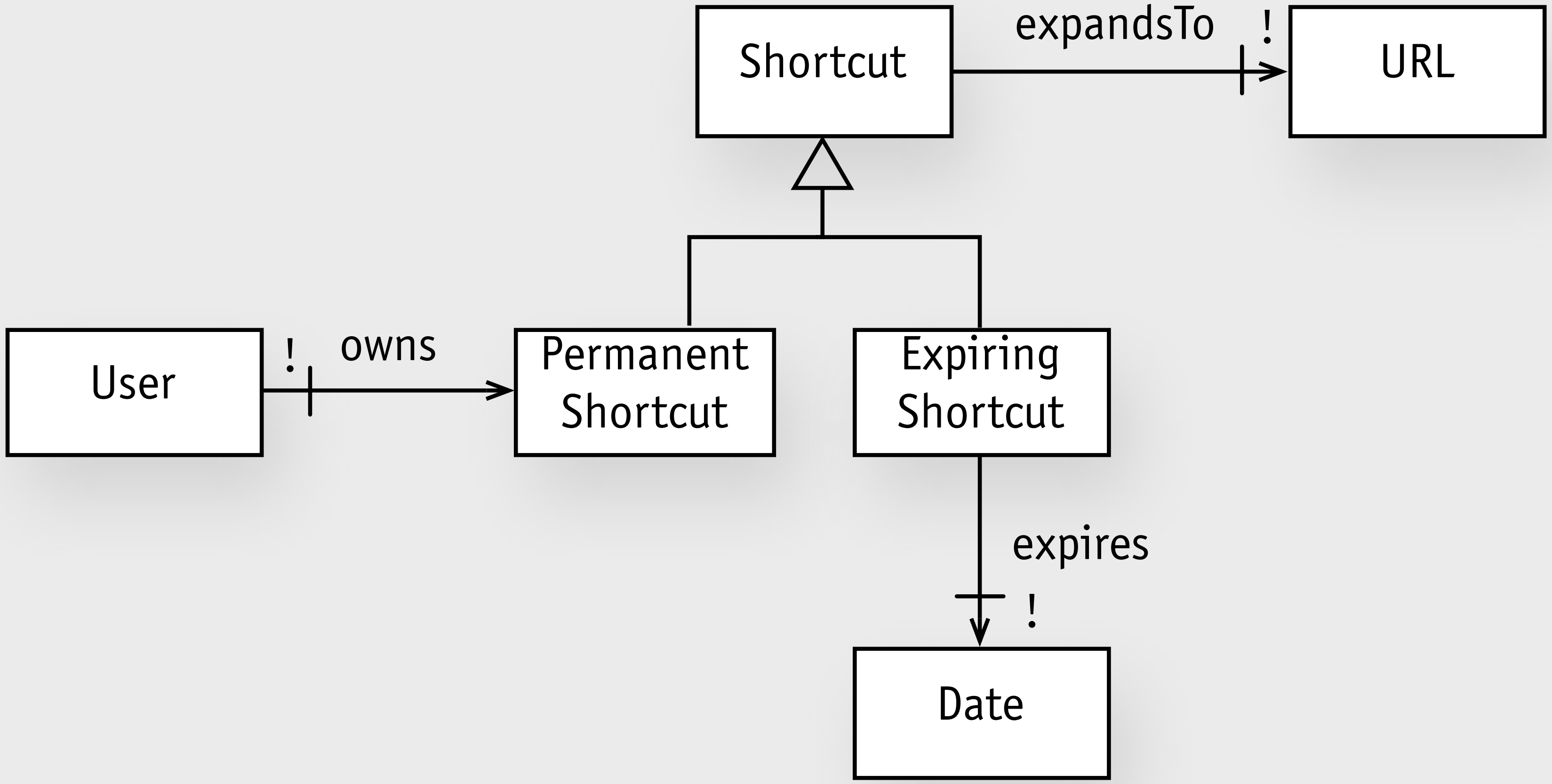
software
studio

data modeling errors

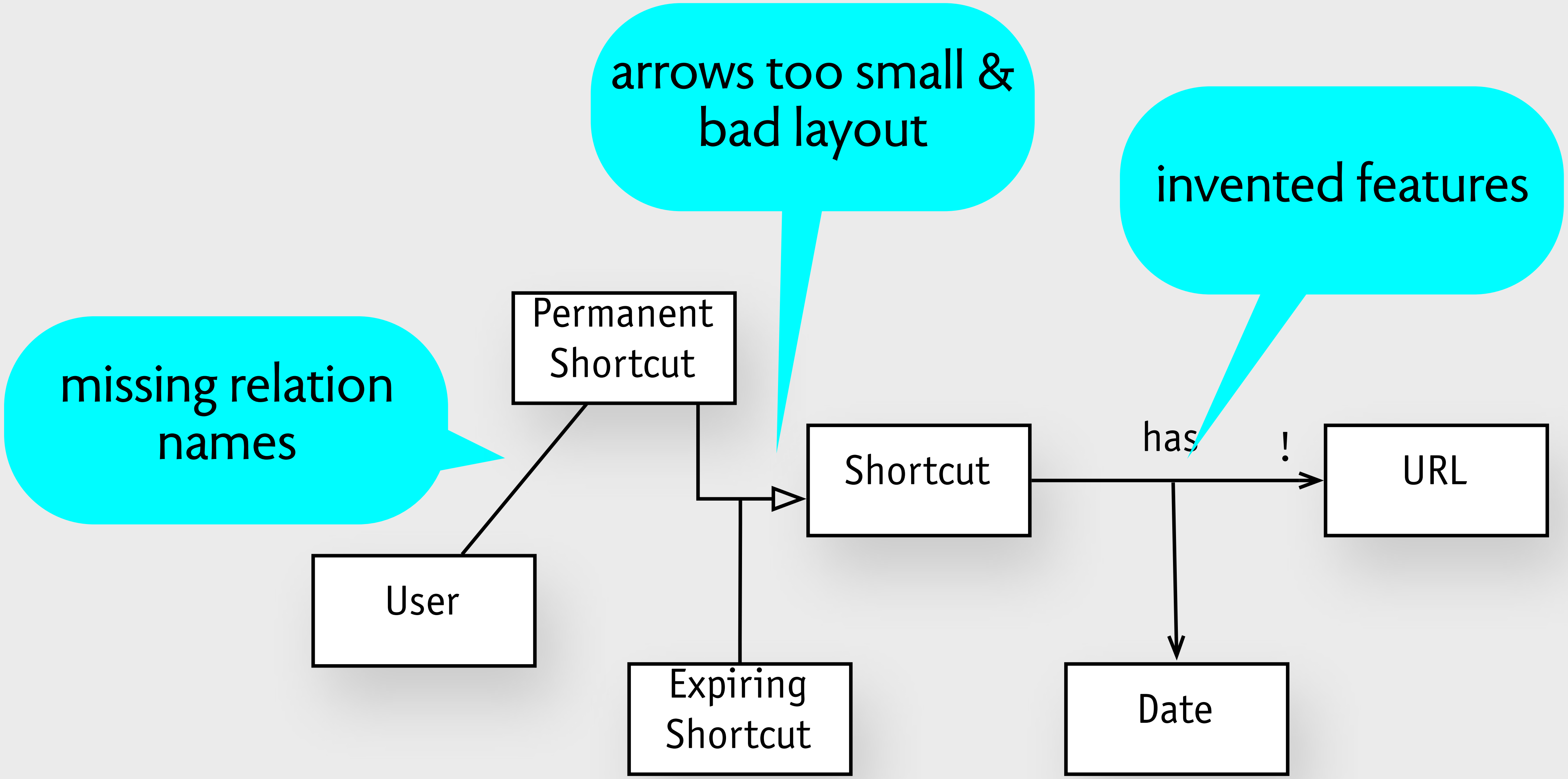
Daniel Jackson and Arvind Satyanarayan

classic errors

suppose this is the data model



bad: not a data model syntactically



arrows too small & bad layout

invented features

missing relation names

Permanent Shortcut

Shortcut

has

!

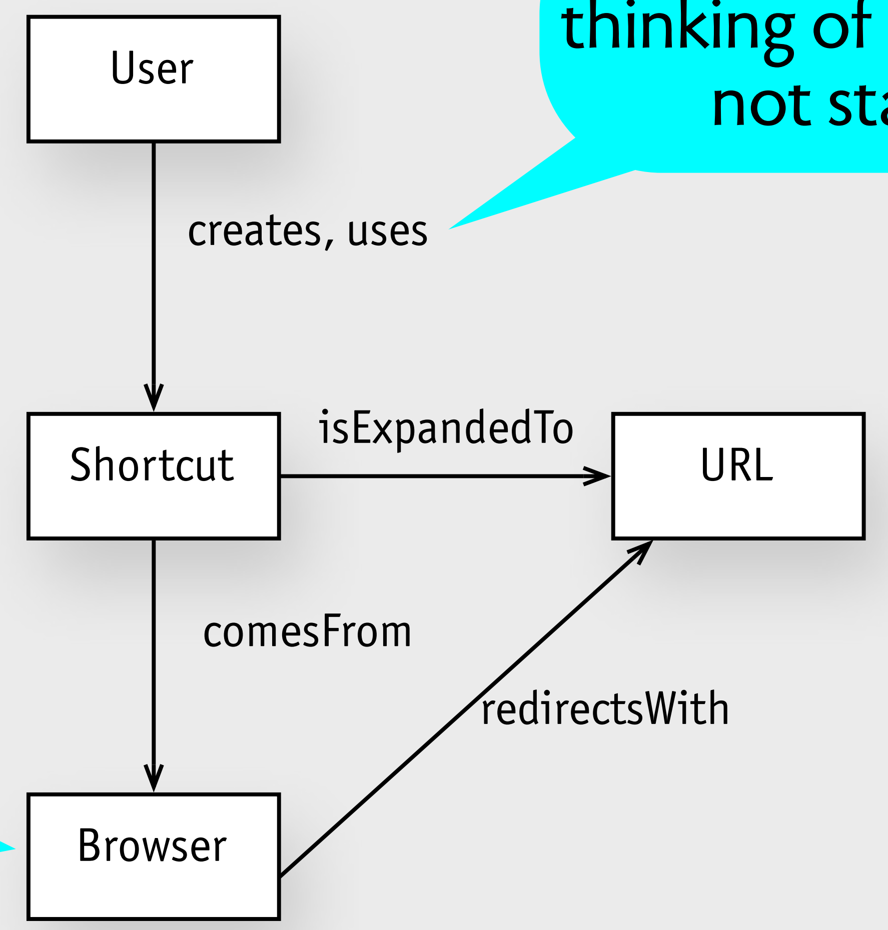
URL

User

Expiring Shortcut

Date

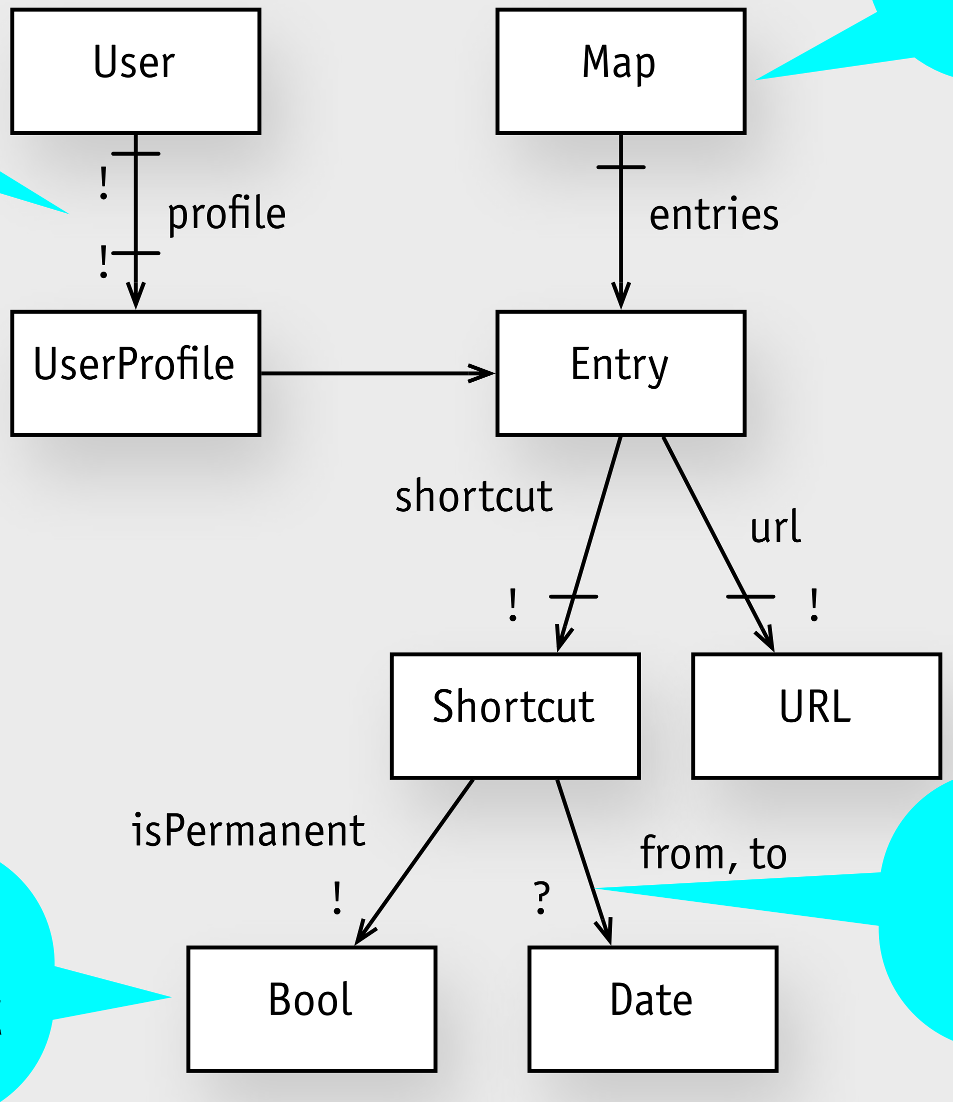
bad: not a data model semantically



designer is thinking of actions, not state

designer thinking of agents, not atoms

bad: implementation details



1:1 relation is a tipoff

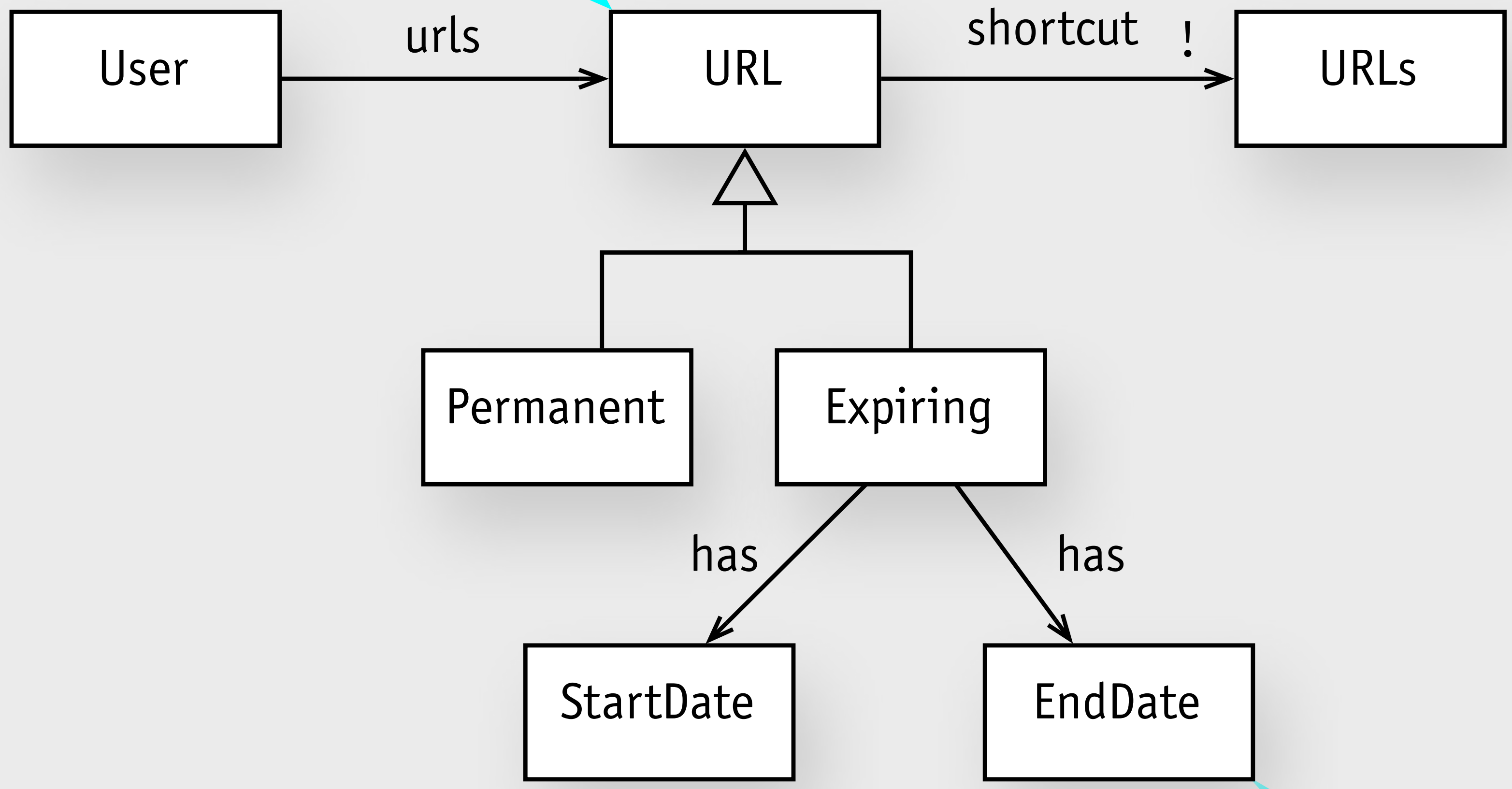
collection types not needed

boolean field: should be subset

leads to weak multiplicities

bad: wrong semantics

relation not subset; classification is per user



User

urls

URL

shortcut

!

URLs

Permanent

Expiring

has

has

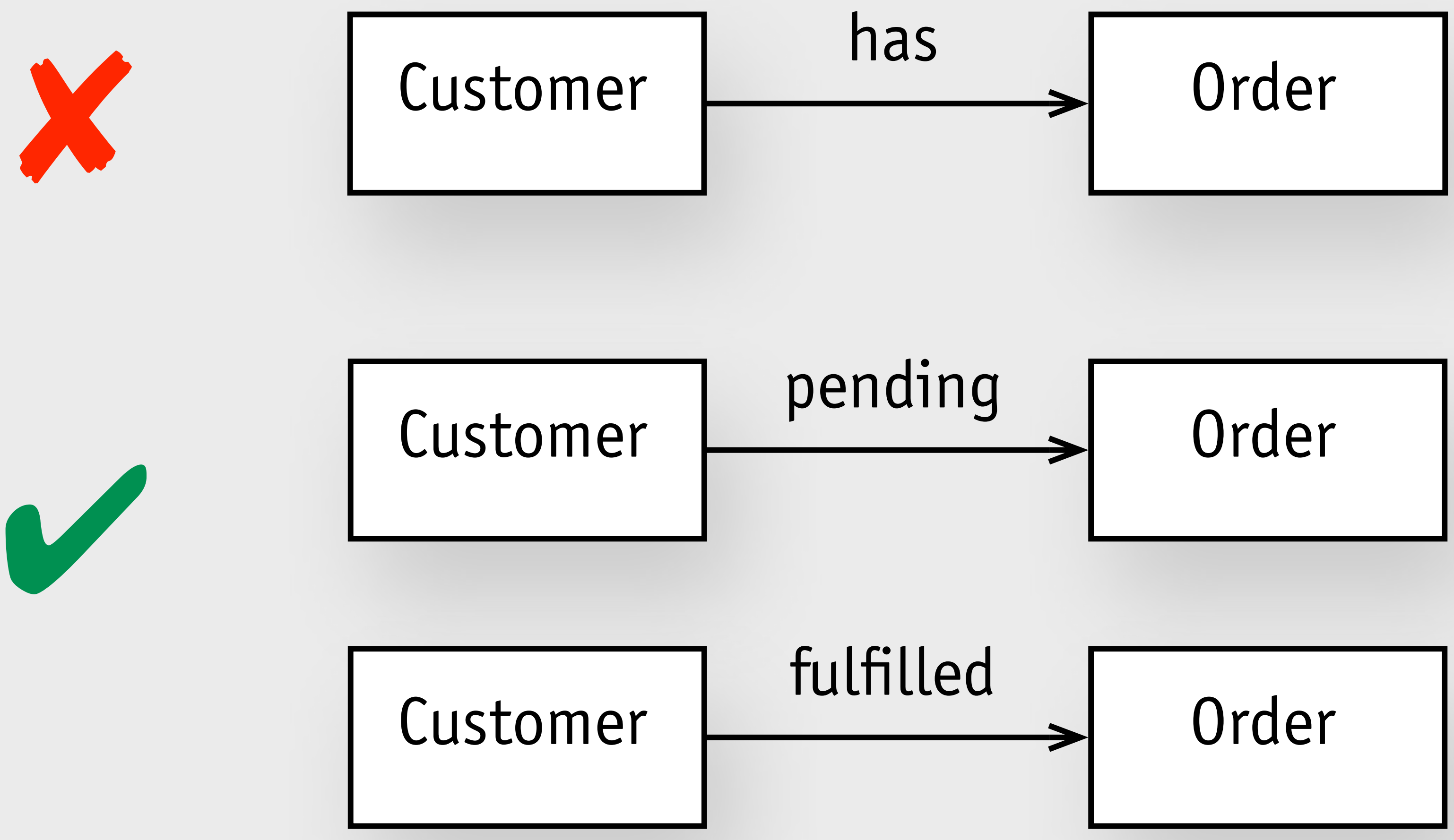
StartDate

EndDate

field names, not sets: disjoint!

bad smells of data modeling

don't be lazy with names



✗

Customer has Order

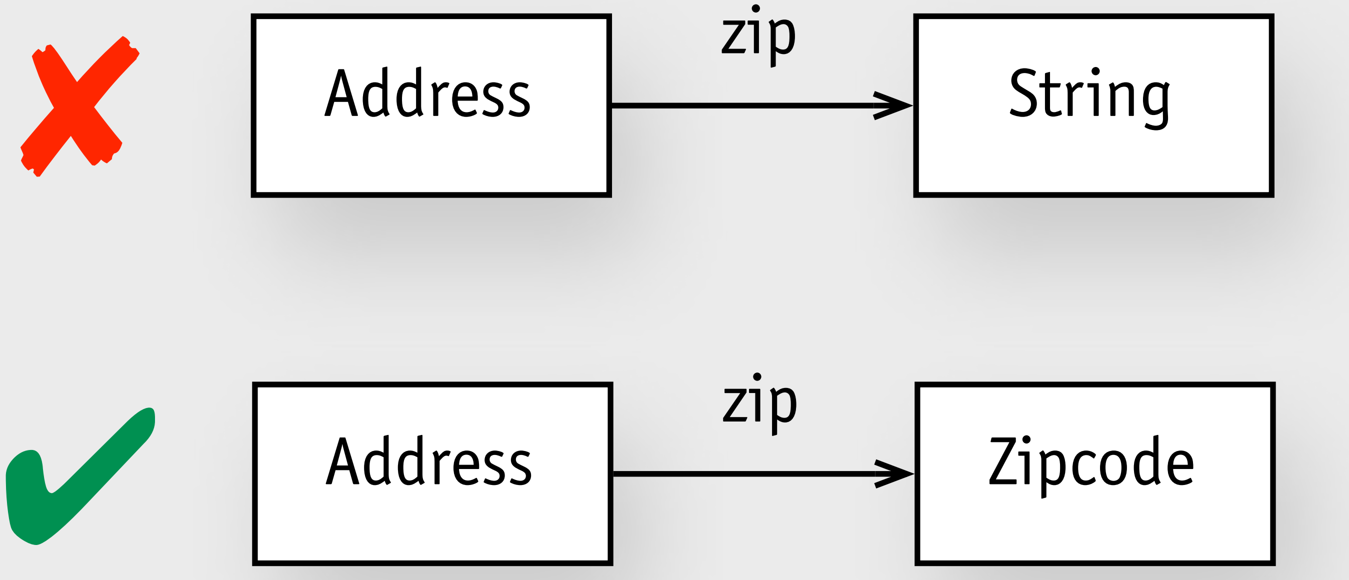
✓

Customer pending Order

Customer fulfilled Order

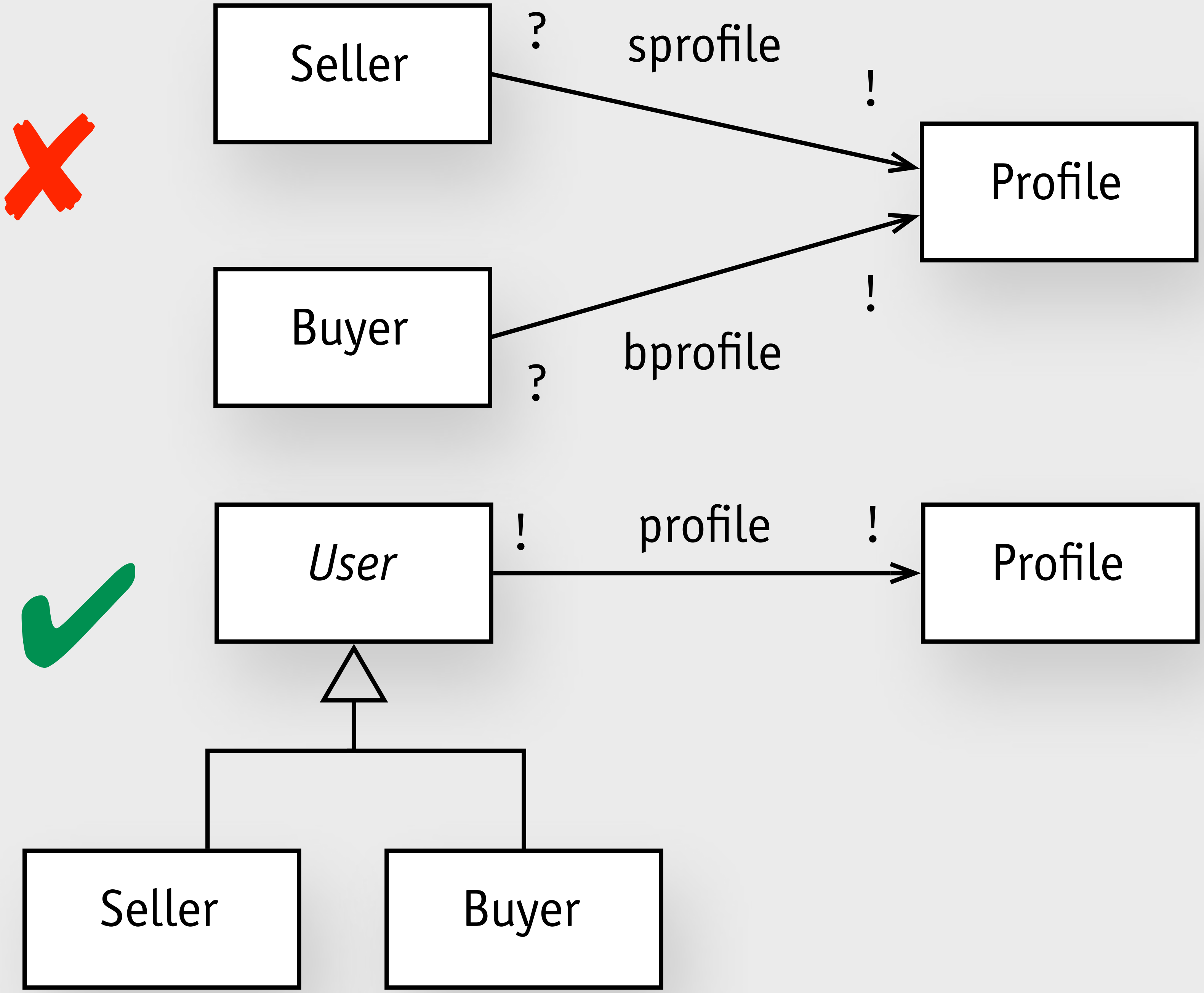
why?
it's never as obvious as it seems
choosing a good name helps designer get clarity

beware primitive types



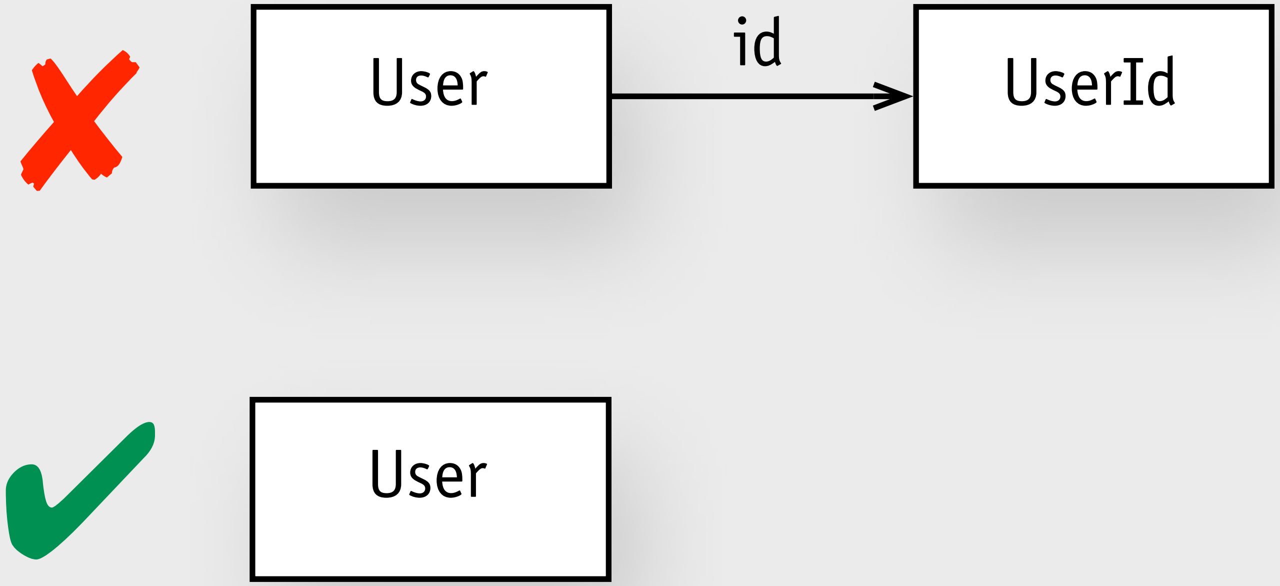
why?
type has syntactic or semantic properties
so may want to store in special way,
and/or use special validators

don’t duplicate



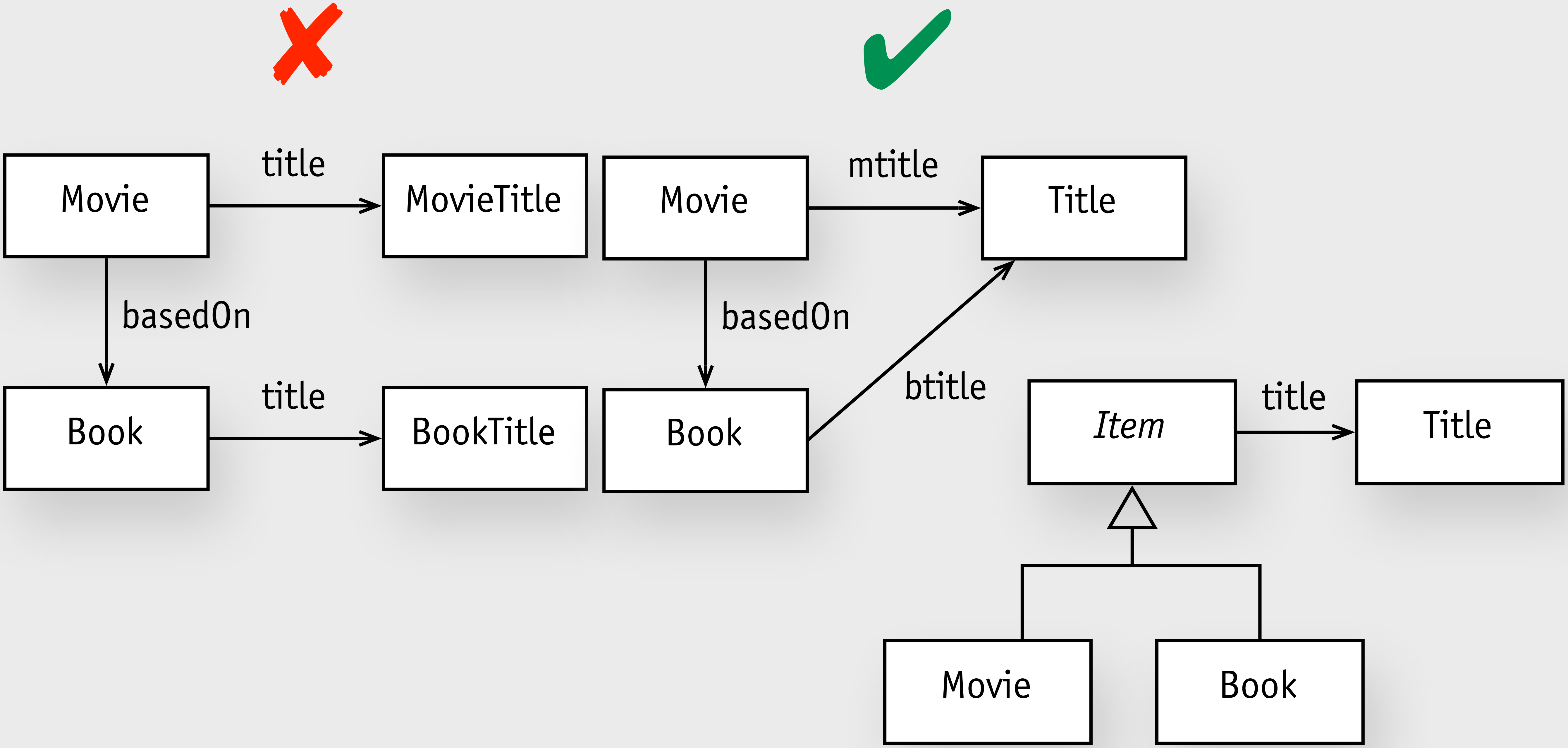
why?
recognize shared properties & generalize
leads to cleaner user interface & cleaner code

don't mention low level ids



why?
every object has an implicit identity anyway
how it's represented is an implementation detail
but note: user-visible ids (such as usernames) *are* relevant

don’t split types

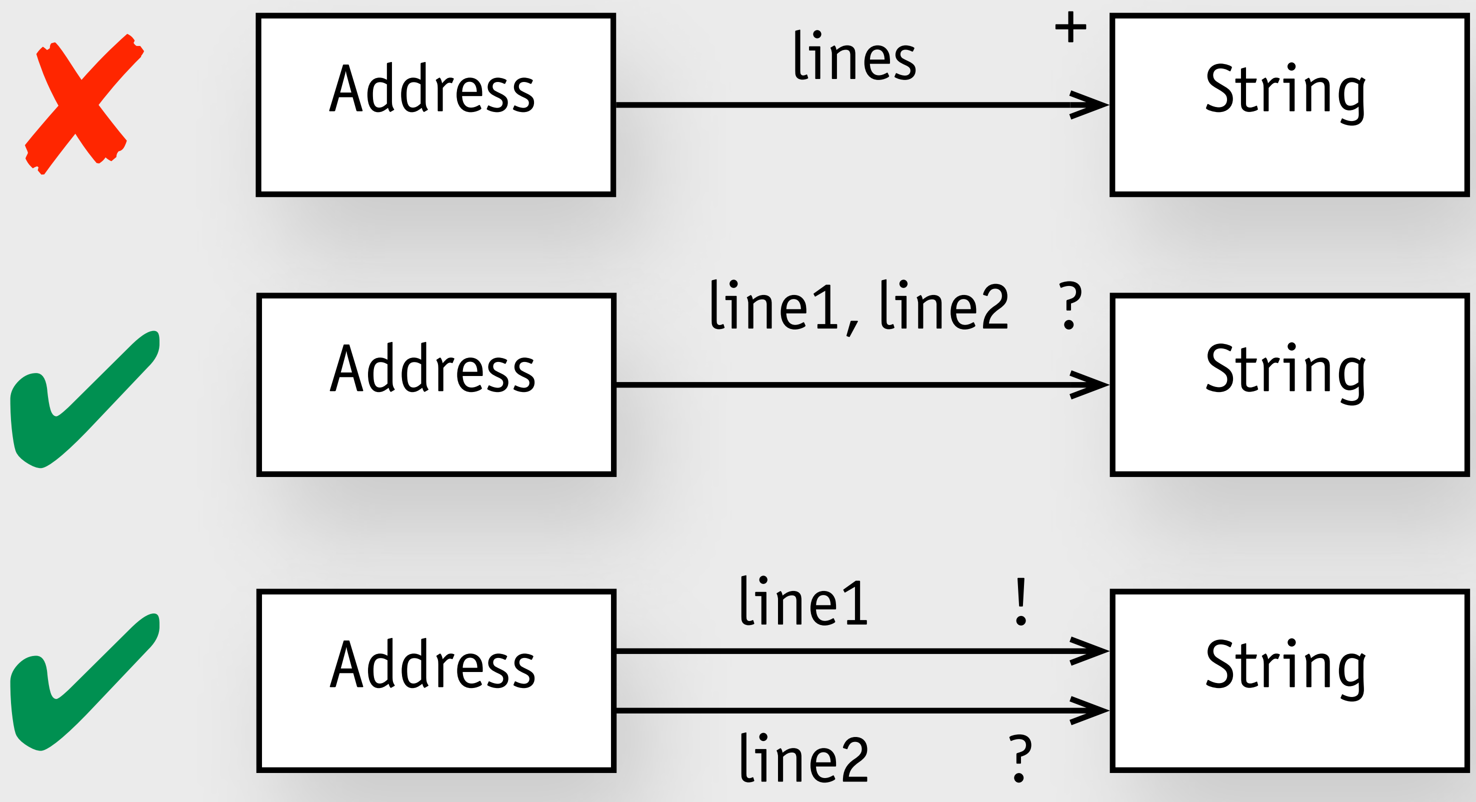


why?

distinct types are disjoint, so couldn’t ask whether movie and book have same title

so atoms to be compared must have the same type

don’t use set when order matters



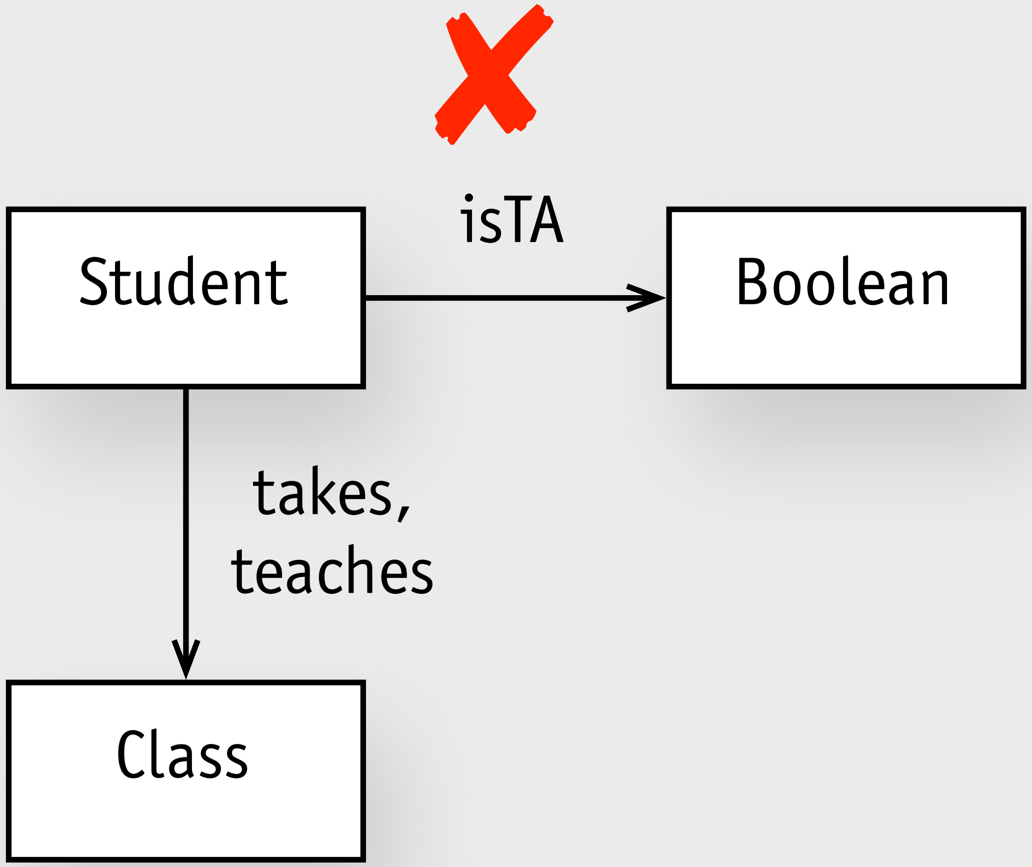
✗ Address lines + String

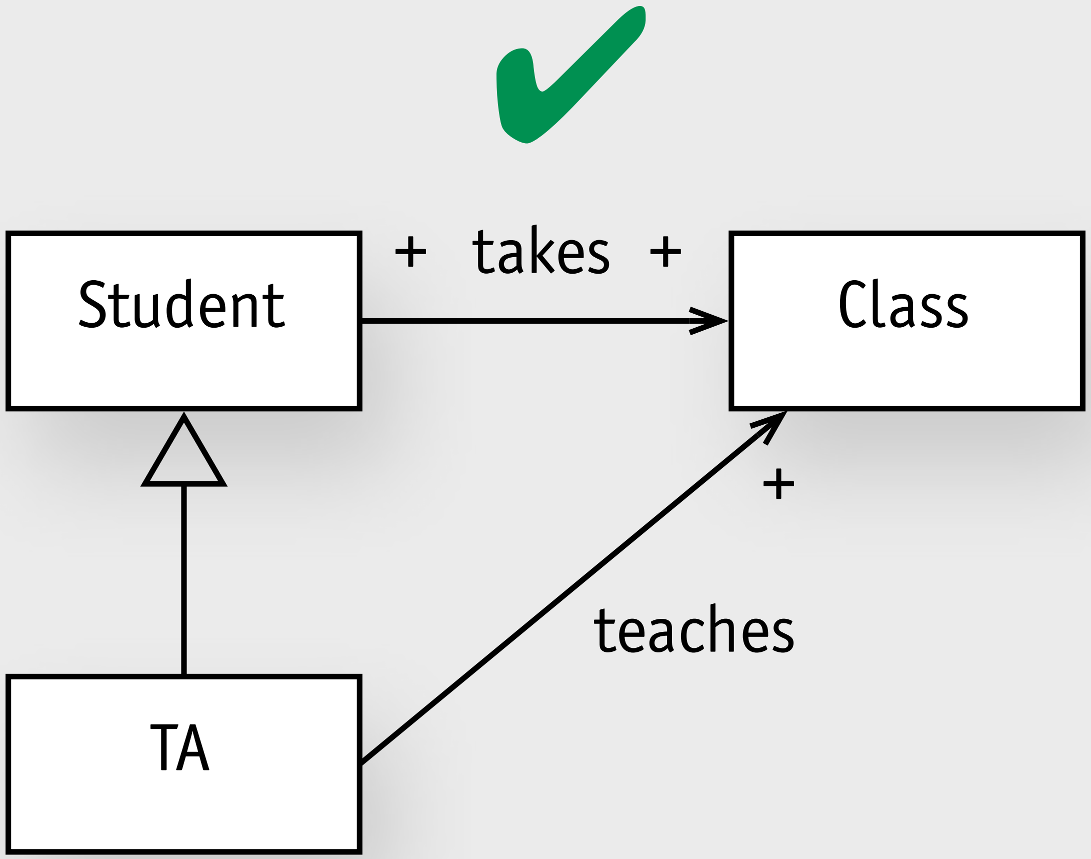
✓ Address line1, line2 ? String

✓ Address line1 ! line2 ? String

why?
tuples of a relation have no order
implementer can choose an unordered collection

use subsets, not boolean flags

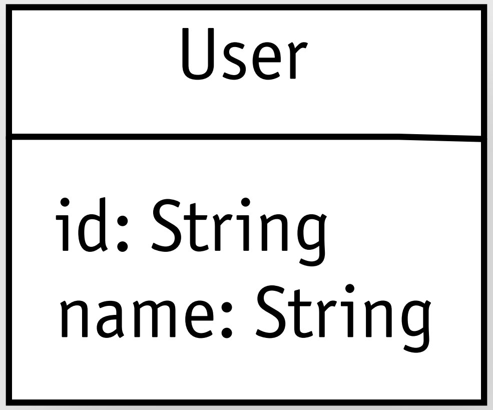

✗
Student
isTA
Boolean
takes,
teaches
Class


✓
Student
+ takes +
Class
+
teaches
TA

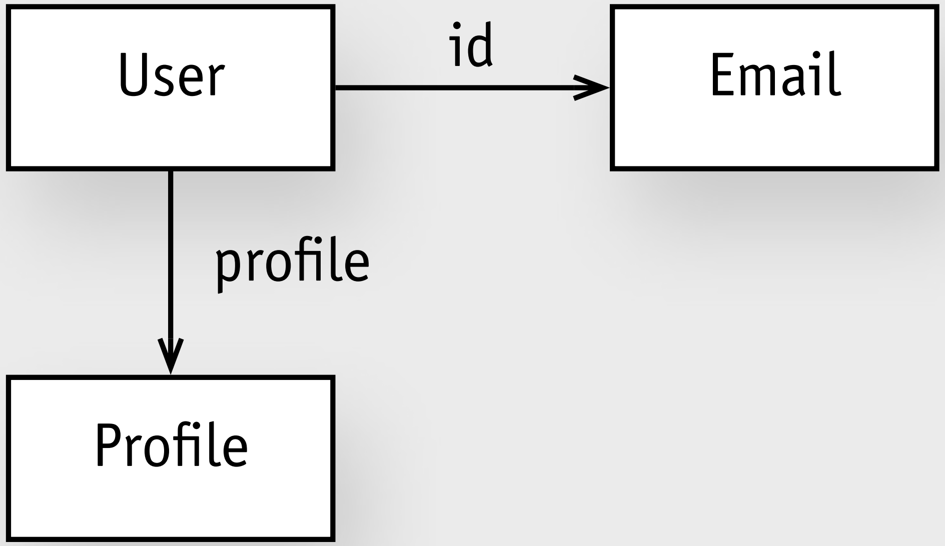
why?
flag is low level way to represent
obscures dynamic classification
prevents recording multiplicity graphically

don’t use attributes

✗

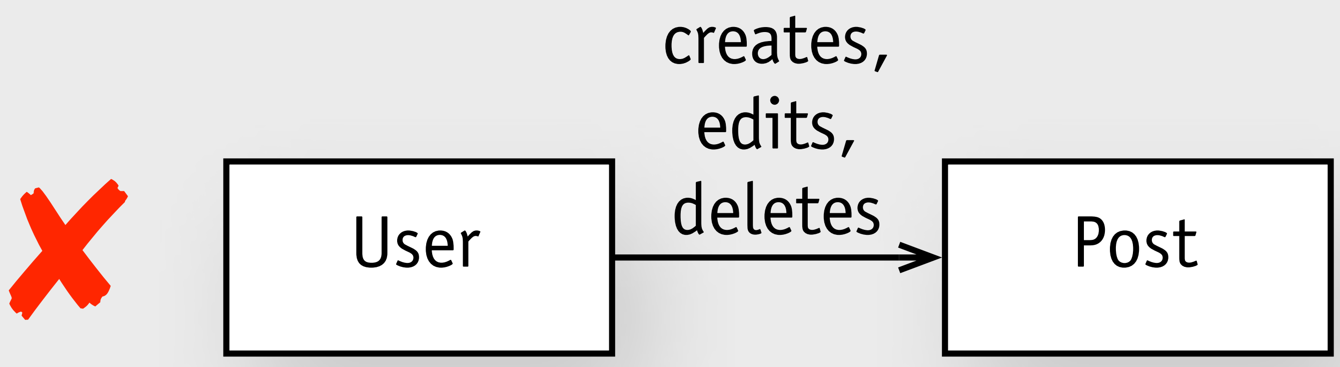


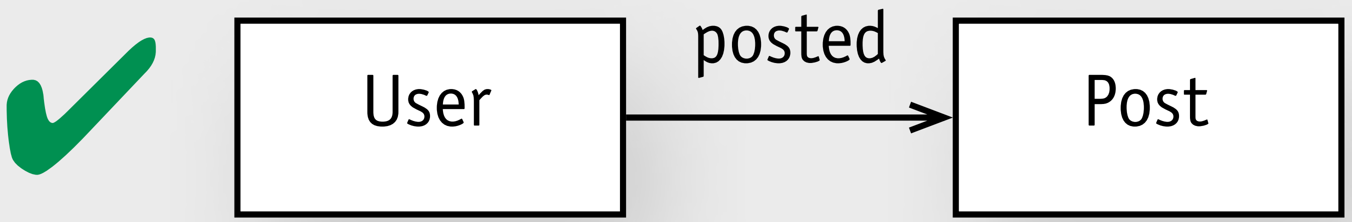
✔



why?
attributes are an ill-defined idea, and just complicate the notation
better to factor out the relations that matter

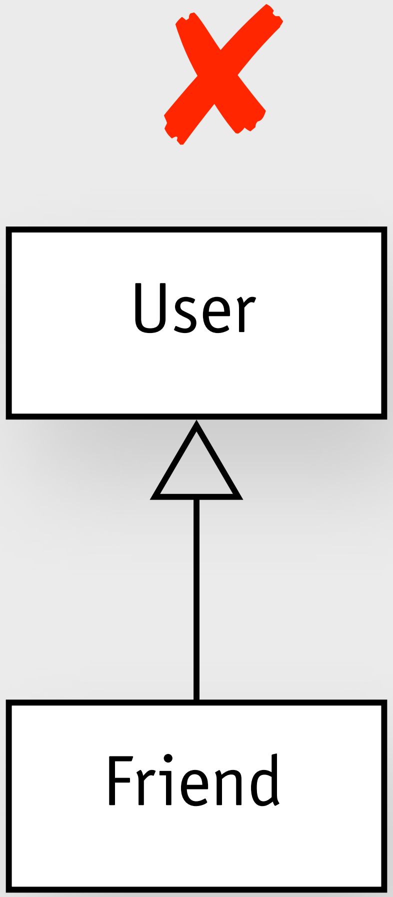
don't confuse state with actions

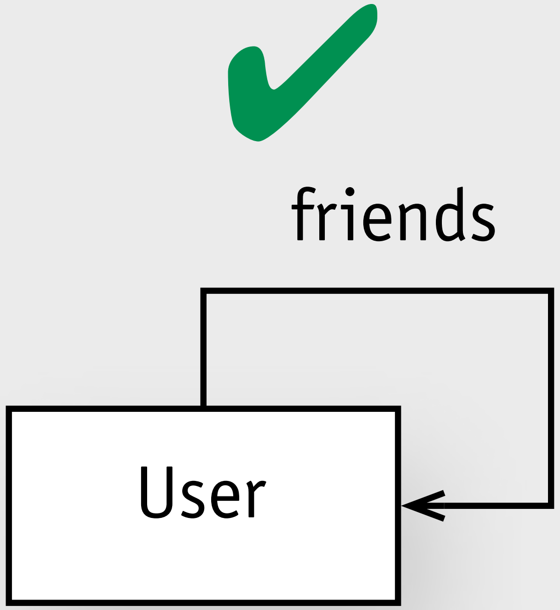




why?
data model describes what is *remembered*
that is, what's stored in the state

don’t use subsets for relational state

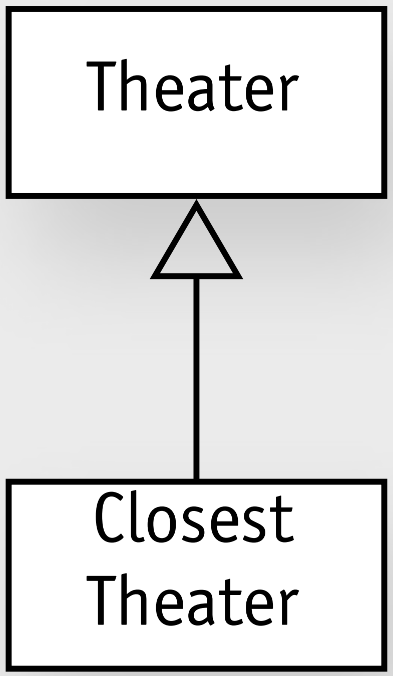




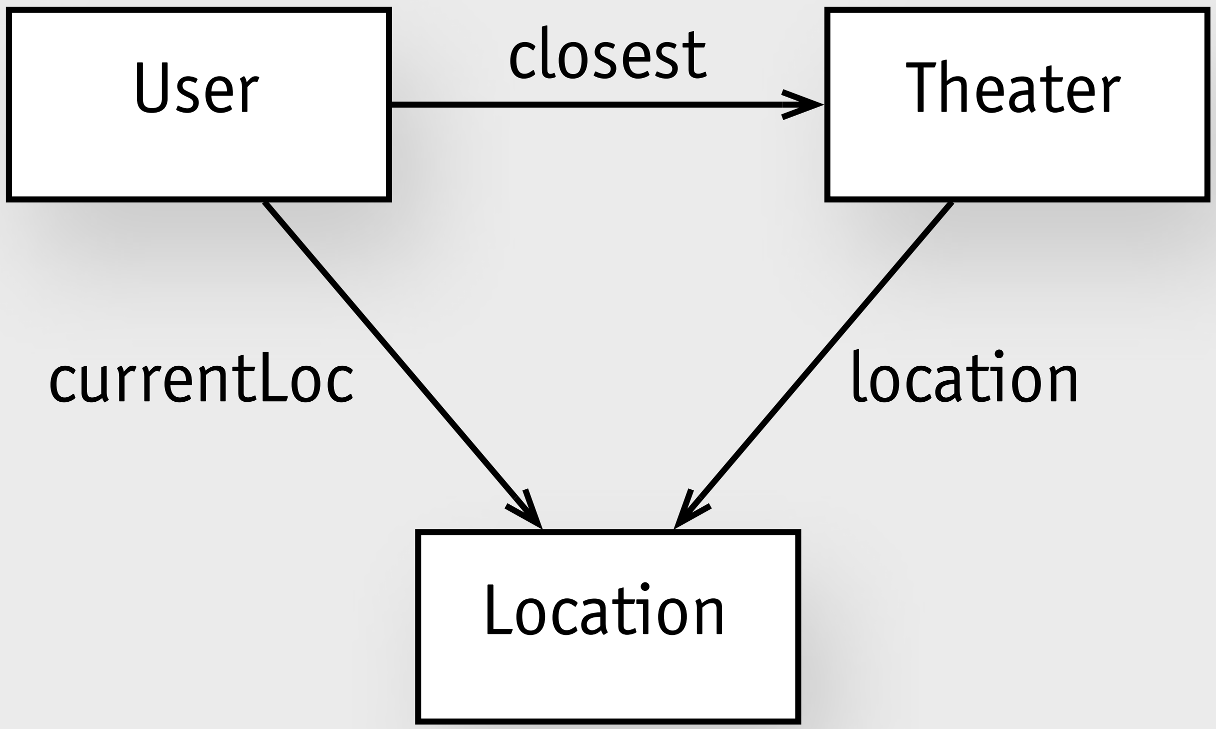
why?
subset is with respect to a context (a user)
without this, data structure won’t work

another example of bad subset

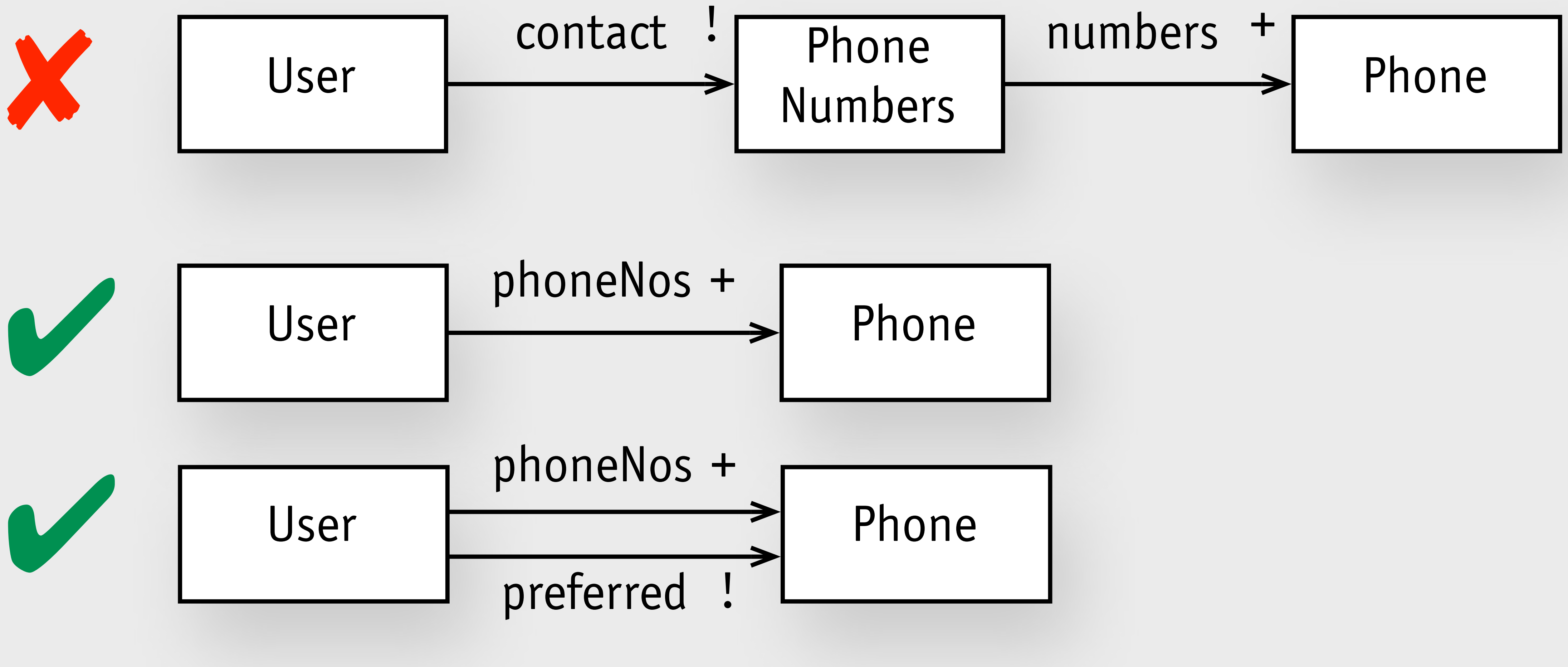
✗



✓

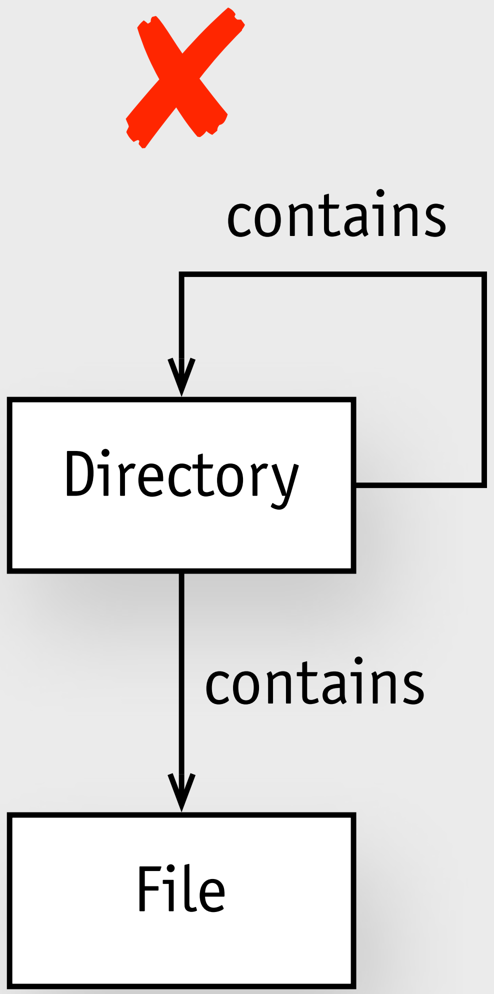


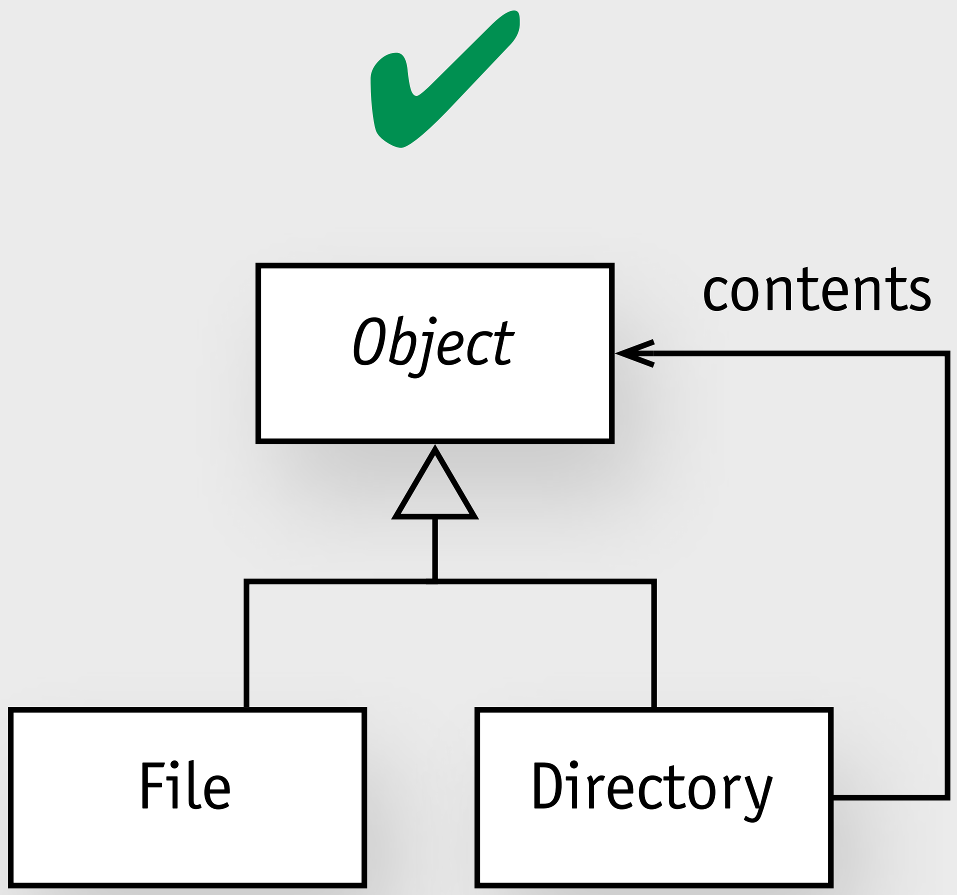
collections aren't domain objects



why?
collection objects are implementation details
unless they have properties beyond their contents

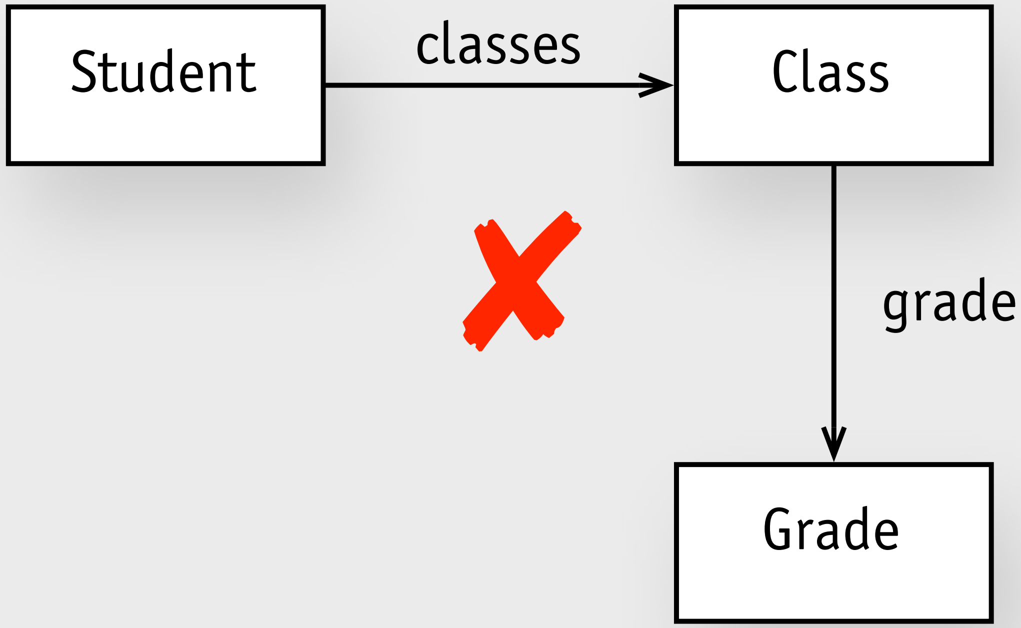
don't split a relation

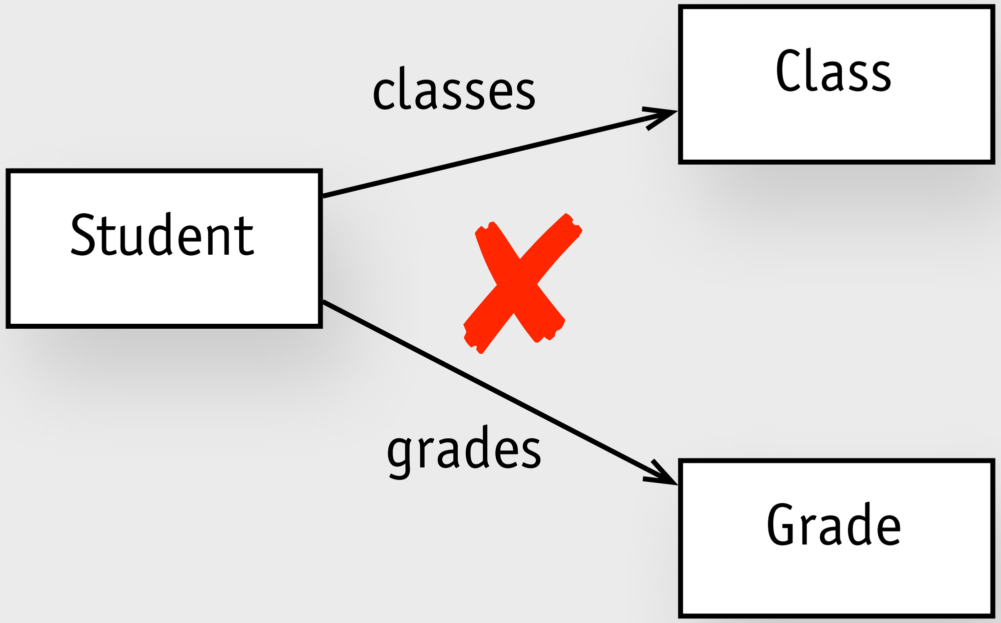


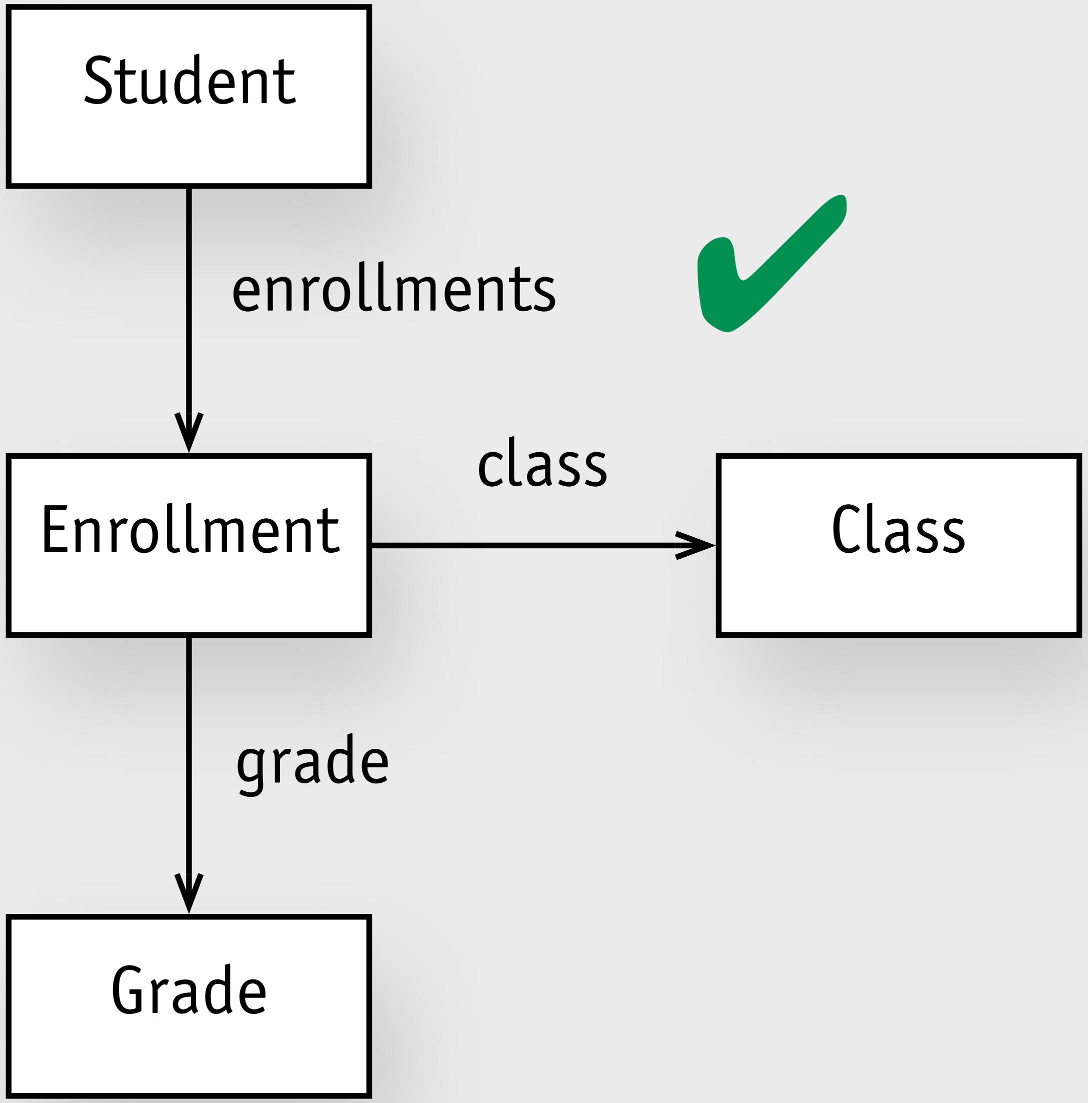


why?
splitting obscures generalization
leads to duplication in code

watch out for 3-way relations



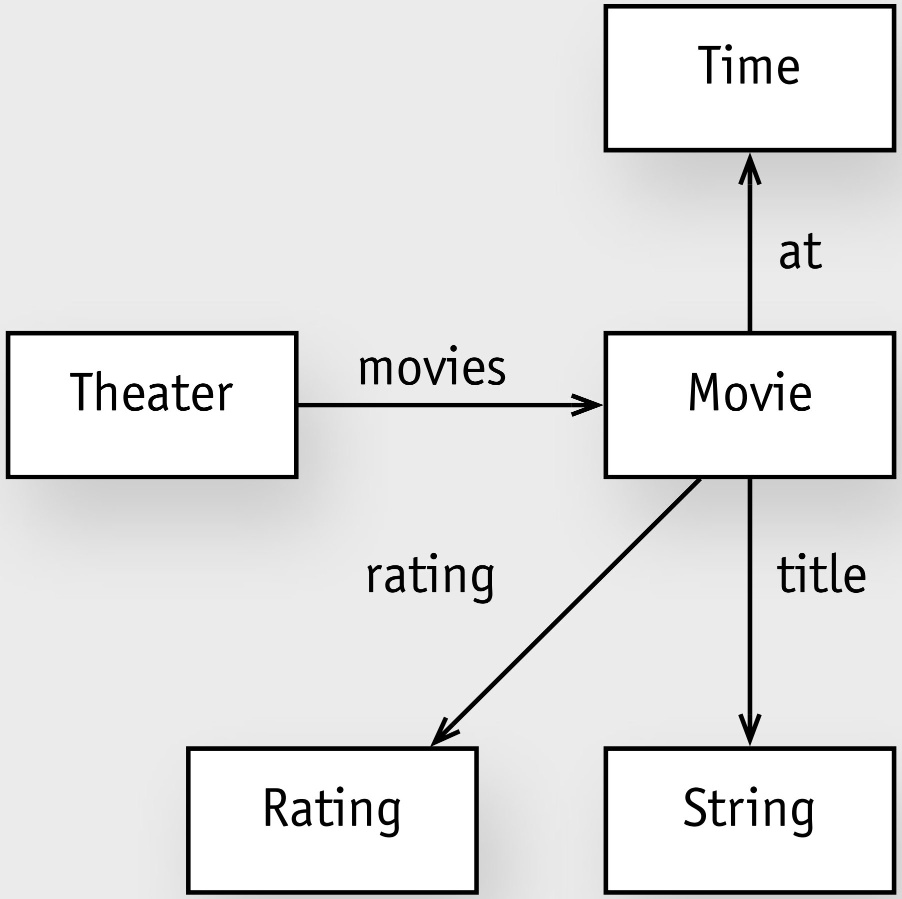




why?
student-class-grade is a 3-way relation
need a “tuple” type such as Enrollment

another example of 3-way relation

✗



Time

at

Theater

movies

Movie

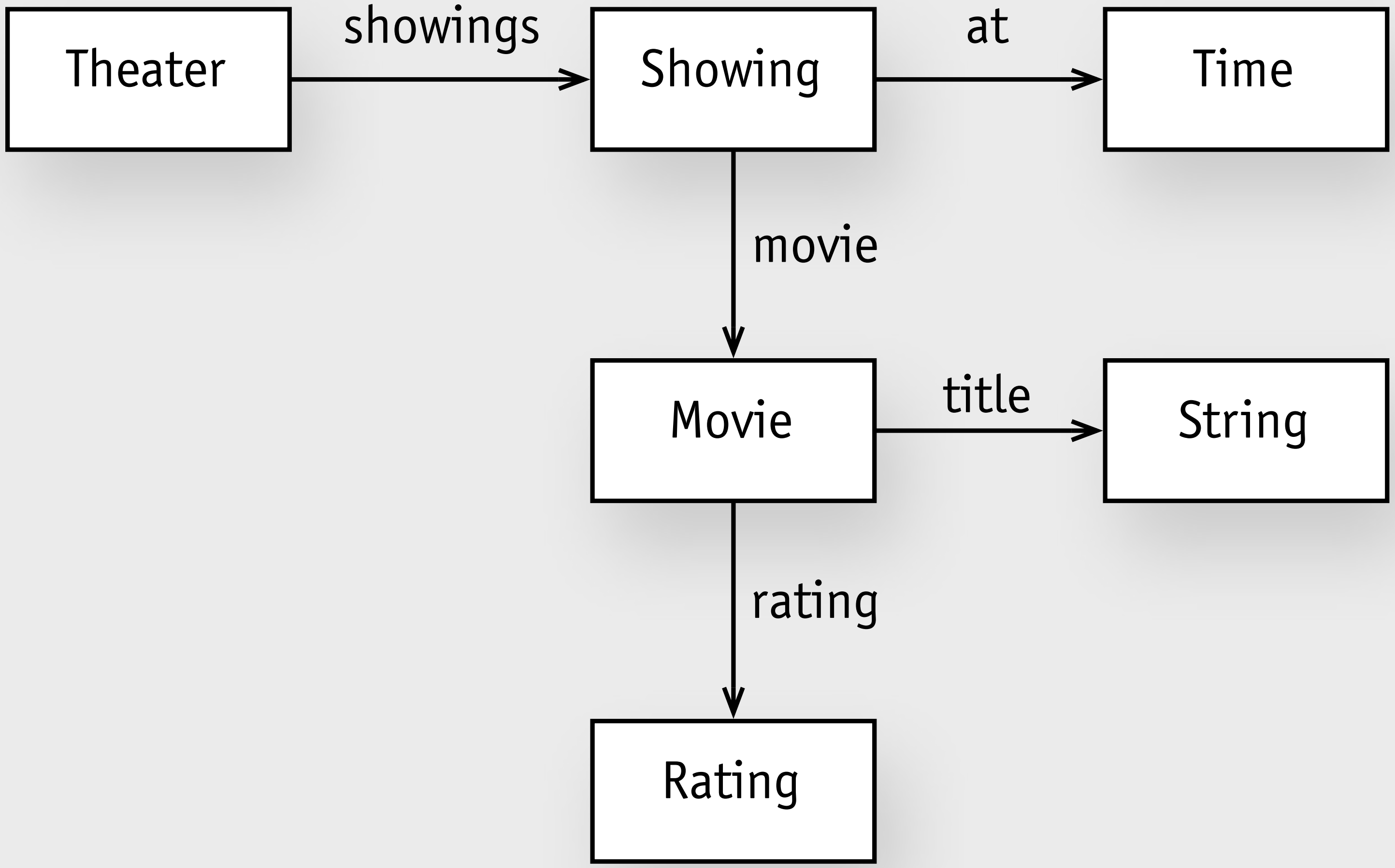
rating

title

Rating

String

✓



Theater

showings

Showing

at

Time

movie

Movie

title

String

rating

Rating

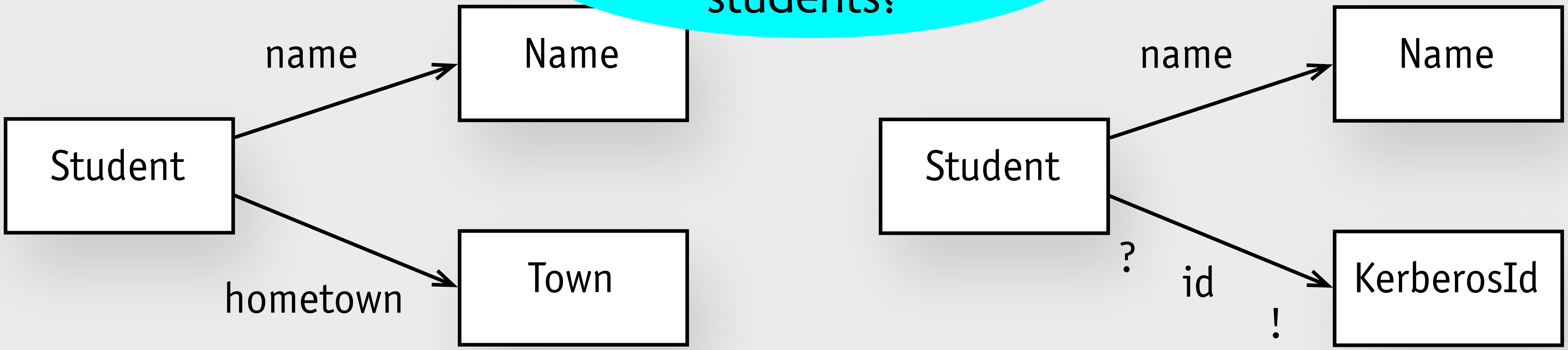
model data that matters

don’t ask ‘what do I know about students?’

✗

but: ‘what must my app know about students?’

✓



why?
no point modeling the easy stuff
focus on the hard parts
in this case how a student is identified matters
home town probably does not (except maybe for MIT Giving)